P11
User's Manual

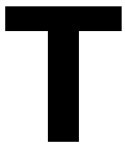
68HC11 Microcontroller
Programming Hardware and Software

Products Described

P11 Programmer Board
P11.EXE DOS Bootloader Program
WP11.EXE Windows Bootloader Program

68HC11 Devices Supported

MC68HC711E20
MC68HC711E9
MC68HC811E2
MC68HC711D3
MC68HC11A1/A8
MC68HC711K4



The Engineers Collaborative, Inc.
Web Site at http://www.tec-i.com
Email support@tec-i.com

* * * IMPORTANT WARRANTY AND LIABILITY INFORMATION * * *

The Engineers Collaborative, Inc. warrants the diskette, if any, on which the software is furnished to be free from defects in materials and workmanship for a period of 90 days from the date of shipment to you. The P11 Programmer hardware is warranted for a period of 6 months. This includes parts and labor and free technical support. Except as indicated above, there are no other warranties, expressed or implied. No warranty of fitness for a particular purpose is offered. The user is advised to test the software thoroughly before relying on it. The user assumes the entire risk of using the product. The total liability of The Engineers Collaborative, Inc. is limited to the purchase price of the product, and does not cover any lost profits, special, incidental or consequential damages, or any claim against the purchaser by any party.

* * * SOFTWARE LICENSE STATEMENT * * *

This software is protected by US copyright law and international treaty provisions. Therefore, you must treat this software *just like a book*, with the following exception. The Engineers Collaborative, Inc. authorizes you to make archival copies of the software for the sole purpose of backing-up our software and protecting your investment from loss. By saying, *just like a book*, The Engineers Collaborative, Inc. means that the software can not be used by two different people in two different places at the same time.

* * * OTHER * * *

The information contained in this manual has been carefully checked and is believed to be accurate and complete at the time of printing. However, no responsibility is assumed for errors that might appear. The Engineers Collaborative, Inc. reserves the right to make changes to the product and/or the manual at any time without notice. Furthermore, The Engineers Collaborative, Inc. assumes no liability arising out of the use or application of any of its products. No part of this document may be copied or reproduced in any form or by any means without prior written consent of The Engineers Collaborative, Inc.

(C) Copyright 1986-2004 The Engineers Collaborative, Inc.
Web Site at www.tec-i.com
Email support@tec-i.com
All Rights Reserved
Printed in the U.S.A.

TABLE OF CONTENTS

1.0 INTRODUCTION 5
2.0 PACKING LIST 6
3.0 SYSTEM REQUIREMENTS 6
3.1 P11.EXE 6
3.2 WP11.EXE 6
4.0 DEVICES SUPPORTED 7
5.0 GETTING STARTED 7
5.1 Software Installation 7
5.2 Initial Test Of The Programmer 9
6.0 THE P11 PROGRAMMER BOARD 10
6.1 Power Requirements 10
6.2 RS-232 Port Connection 11
6.3 Reset Push-Button Switch 12
6.4 R1 Vpp Adjustment 12
6.5 J1 Vpp Control Jumper 12
6.6 J2 Reset Hold Jumper 13
6.7 J3 and J4 Mode Jumpers 13
6.8 ZIF Sockets 14
7.0 THE WP11.EXE PROGRAM 15
7.1 Program Overview 15
7.2 WP11 Screen Shot 15
7.2.1 WP11 Status Panel 16
7.2.2 Changing the Config Register 16
7.3 WP11 Menus 16
7.3.1 File Menu 16
Load Buffer from File 16
Load Buffer from Chip 17
Save Buffer 17
Save Buffer As 17
Exit Program 17
7.3.2 Edit Menu 17
Clear Buffer 17
Fill Buffer 18
Verify Buffer Against Chip 18
Edit Buffer Contents 18
7.3.3 Select Device Menu 19
7.3.4 Communications Menu 20
Com Port Setup and Test 20
Com Port Troubleshooting Chart 21
7.3.5 Options Menu 22
File Format Options 22
Load Buffer On Startup 22
Turn Off Hint Boxes 22
Turn Off Status Line Hints 23
Turn Off Memory Map Checks 23
7.3.6 Help Menu 23
Contents 23
About 23
7.4 Chip Operation Buttons 23
Initialize Device 23
Communications Check 24
Blank Check 24
Erase EEPROM and Config Reg 24

Program Config Reg 24
Program EEPROM 24
Program EPROM 25
Program Entire Device 25
8.0 TUTORIAL EXAMPLE SESSION 25
9.0 USING THE P11.EXE PROGRAM 27
9.1 Invoking P11.EXE 27
9.1.1 Command Line Options 27
9.1.2 Command Line Only Options 28
9.2 P11 Screen Display 29
9.3 P11.EXE Menu Options 29
Change Device Type 29
Change Buffer Fill Character 30
Change config Register 30
Load Object Code File to Buffer 30
Device Menu 31
Initialize Device 31
Communications Test 31
Blank Check Device 31
Bulk Erase EEPROM 32
Read Device Contents into Buffer 32
Verify Device Contents against Buffer 32
Program &Verify Config Register 32
Program &Verify EEPROM 32
Program &Verify EPROM 32
Program & Verify Entire Device 32
Exit to Main Menu 32
Examine/Edit Buffer Contents 32
Save Buffer in a File 33
Change communications Port 33
Help 33
Exit to DOS 33
APPENDIX A P11 Programmer Board Schematic Diagram 34

1.0 INTRODUCTION

This manual provides information on three separate but related products that are used for programming 68HC11 family microcontrollers. These products are:

1) The P11 programmer board.
2) P11.EXE a DOS program written for the P11 programmer board or any circuit that supports the 68HC11 bootstrap mode of programming.
3) WP11.EXE a Windows program written for the P11 programmer board or any circuit that supports the 68HC11 bootstrap mode of programming.

These products are used to program the on chip memory of the Motorola 68HC11 series microcontrollers. Depending on the product you may have received all three products or just one of the software programs.

The P11 programming board is designed to be attached to the serial port of an IBM PC or compatible computer.

The P11.EXE and WP11.EXE software takes input from object code files created with HC11 assemblers or compilers. These object code files may be in Motorola S-record, Intel Hex or binary memory image formats. The software also allows the operator to perform many other useful programming functions such as erasing, blank checking & verifying devices, displaying, editing, exporting and changing the format of the object code files and filling unused memory locations with a user specified "fill" byte. The software was designed to work with the *TECI* P11 Programming Board but will also work with any hardware that supports the HC11 special bootstrap mode of operation.

TECI provides several other development tools for 68HC11 family microcontrollers such as our Cross Assemblers WASM11.EXE and TASM11.EXE and real time in-circuit emulator TECICE-HC11.

2.0 PACKING LIST

*** * * Important Note * * ***

This section applies only if you purchased the P11 programming board.
You should find these items in the shipping carton:

1. Manual

2. Software Diskette
   Files on the Distribution Diskette:
   `P11SETUP.EXE`

3. P11 Programmer Board (if purchased with the software)

4. A wall mounted transformer for powering the P11 Programmer Board

3.0 SYSTEM REQUIREMENTS

3.1 P11.EXE

P11.EXE requires an IBM PC or compatible with at least 512K RAM memory, one 5.25" or 3.5" disk drive, one serial port and PC-DOS Version 2.1 or higher or the equivalent MS-DOS version operating system.
P11.EXE may be run from Microsoft Windows 3.1, 98, ME, NT in full screen text mode or in a DOS window. Read your Windows manual for more information about running DOS programs from Windows.

*** * * Important Note * * ***

P11.EXE ***does not*** work with Windows 2000 or WINDOWS XP. For these operating systems you must use WP11.EXE.

The P11 Programmer Board is designed to be connected to a serial (RS-232) port of the IBM family of personal computers, designated either COM1 or COM2.

3.2 WP11.EXE

WP11.EXE requires a minimally configured IBM PC or compatible with at least 16Meg RAM memory, one 3.5" disk drive, one free serial port and Windows 3.1, 98, ME, NT, 2000 or XP.

4.0 DEVICES SUPPORTED

Device	Package				
	48 Pin DIP	52 Pin PLCC	40 Pin DIP	44 Pin PLCC	84 Pin PLCC
MC68HC711E9	N/A	YES	N/A	N/A	N/A
MC68HC711E20	N/A	YES	N/A	N/A	N/A
MC68HC811E2	YES	YES	N/A	N/A	N/A
MC68HC711D3	N/A	N/A	YES	YES	N/A
MC68HC11A1/A8	YES	YES	N/A	N/A	N/A
MC68HC711K4	N/A	N/A	N/A	N/A	YES

In the future, more devices will be supported.

5.0 GETTING STARTED

5.1 Software Installation

The software is contained in a single self extracting .EXE file called p11setup.exe. This is a standard windows install file and you need to be running windows for it to guide you through the install process. You may have downloaded the setup.exe file from our web site at www.tec-i.com or received the file on a single 3.5" floppy diskette. To start the setup process you can either double click on the file name from windows explorer or use "RUN" from the windows start menu.

The files extracted from setup.exe are:

`UNINSTALL.EXE` Used to uninstall P11. Double click on this file from within Windows Explorer to uninstall.

`INSTALL.LOG` Used by `UNINSTALL.EXE` to uninstall P11.

`P11.EXE` This is the DOS PC program used to control the programmer hardware.

`P11.HLP` This is a standard DOS text file that contains the help screen information for `P11.EXE`. This file can be edited to include additional notes the user may want to have available on-line when using the `P11.EXE` program.

`WP11.EXE` This is the Windows program used to control the programmer hardware.

`WP11.HLP` This is the help file for WP11.EXE.

`P11Manual.PDF` This manual in Adobe Acrobat .PDF format. The Adobe Acrobat reader software is available free of charge from a number of online sites such as: www.adobe.com/products/acrobat/readstep.html

`P11_A8.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711A1/A8 devices. `WP11.EXE` downloads this code to the programmer when A1 or A8 devices are being programmed.

`P11_D3.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711D3 devices. `WP11.EXE` downloads this code to the programmer when D3 devices are being programmed.

`P11_E2.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC811E2 devices. `WP11.EXE` downloads this code to the programmer when E2 devices are being programmed.

`P11_E9.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E9 devices. `WP11.EXE` downloads this code to the programmer when E9 devices are being programmed.

`P11_E20.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711E20 devices. `WP11.EXE` downloads this code to the programmer when E20 devices are being programmed.

`P11_K4.MIK` This is an HC11 object code file in Motorola S-Record format that contains the programming code for MC68HC711K4 devices. `WP11.EXE` downloads this code to the programmer when K4 devices are being programmed.

`TEST_E9.S19` This is an HC11 object code file in Motorola S-Record format. It is used to demonstrate and test the programmer hardware for MC68HC711E9 devices. This file *only* programs the E9 microcontroller's EEPROM memory.

`TEST_E2.S19` This is an HC11 object code file in Motorola S-Record format. It is used to demonstrate and test the programmer hardware for MC68HC811E2 devices. This file *only* programs the E2 microcontroller's EEPROM memory.

`TEST_K4.S19` This is an HC11 object code file in Motorola S-Record format. It is used to demonstrate and test the programmer hardware for MC68HC711K4 devices. This file *only* programs the K4 microcontroller's EEPROM memory.

***** Important Note for DOS users *****

For the DOS software, P11.EXE to work properly it is essential that the directory containing the TECI software be added to the `PATH` in the `AUTOEXEC.BAT` file in the `ROOT` directory on the hard disk. This will enable the user to start up `P11.EXE` from any directory. To enable the `PATH` to the `P11.EXE` the computer must be rebooted after changing the `AUTOEXEC.BAT file`. (Read your DOS manual for more information about the `PATH` command.)

5.2 Initial Test Of The Programmer

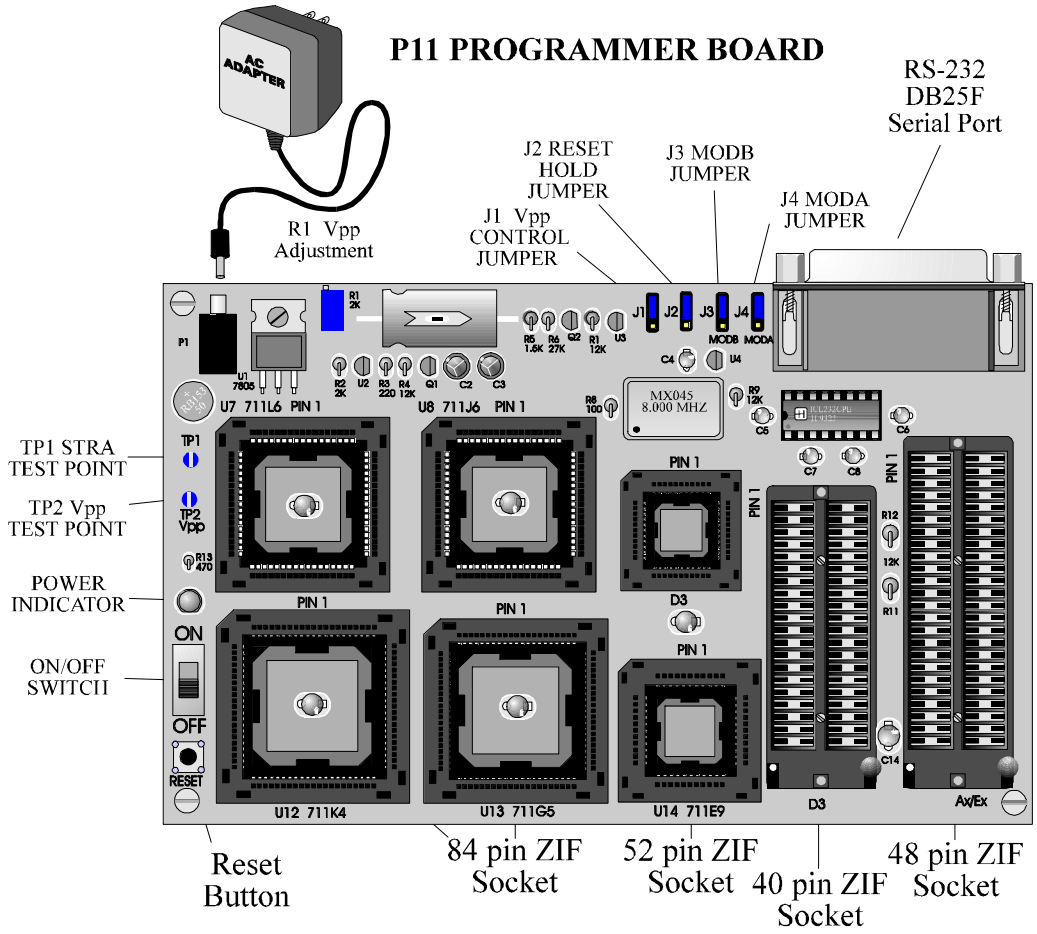
The goal of this section is to verify that the programmer is working properly. To do this we will connect the programmer to the PC and verify that the PC and programmer can talk over a serial connection. The details of how to use the various programming functions and options will come in later sections of the manual. Right now all we want to do is make sure that the programmer and connection to the PC is good.

Follow these steps exactly:

1) Invoke the WP11.EXE program.
2) From the Select Device menu choose a chip that you want to work with.
3) With programmer power turned off, install a chip of the chosen type into the socket specified on the status panel.
4) Referring to section "6.2 RS-232 Port Connection", connect the programmer to the PC with the appropriate cable.
5) Turn power to the programmer on.
6) Press the "Initialize Device" button and follow the instructions given. If you don't get an error message then the "Initialize Device" function worked properly. Some of the other buttons will have become enabled. If this happens then the programmer and the connection are good and you should proceed with reading the rest of this manual. If you received an error message then you must make corrections until you can get the "Initialize Device" function to work error free. By far, the most likely problem is with your serial connection to the PC. Read and perform the tests described in section **7.3.4** of this manual and use the troubleshooting chart provided there to find and fix the problem.

6.0 THE P11 PROGRAMMER BOARD

The P11 programmer is shown below. The following paragraphs describe its major features. A schematic diagram is attached at the end of this document.



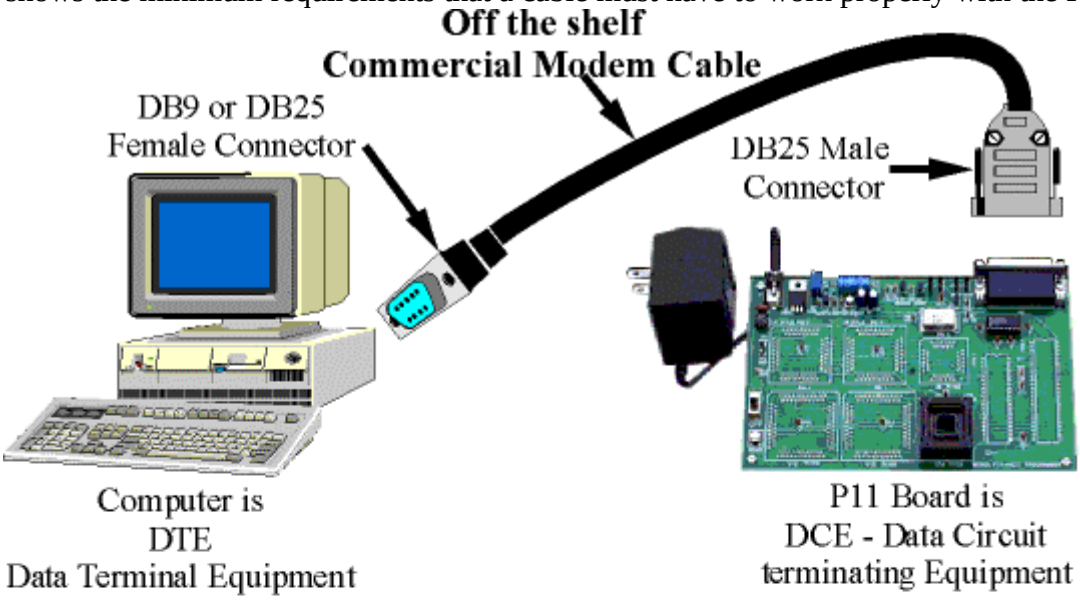
Note: This board shows all the ZIF sockets in place. The version that is for sale will have the socket or sockets requested by the purchaser. The price list is for a board with a single socket. Additional sockets may be purchased from TECI.

6.1 Power Requirements

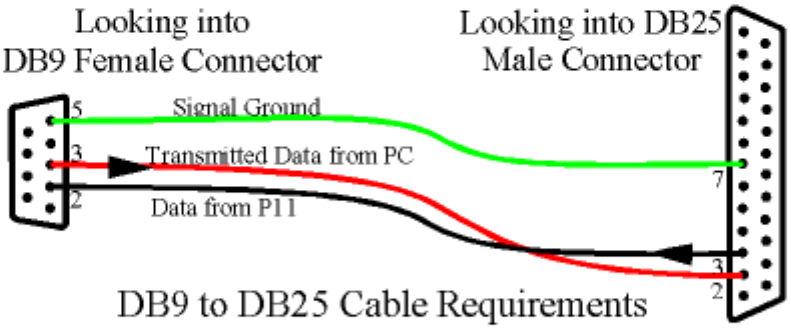
The P11 board requires 12 volts AC about 0.2 Amps. All other voltages for programming 68HC11 chips are generated on the P11 board from this supply. A 12 VAC 1 Amp wall mounted transformer is provided for this purpose.

6.2 RS-232 Port Connection

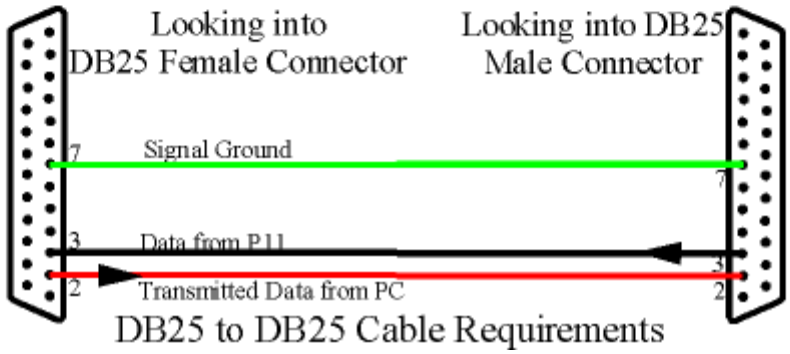
The P11 Programmer connects to the PC via a "Off the shelf" commercial serial cable. The cable must have a DB-9 or DB-25 female connector that matchs your PC on one end and a DB-25 male connector on the other end to mate with the P11 board. Standard modem cables work fine. TECI does not supply this cable. Communications with the P11 does not require all of the wires in a commercial cable, although the presense of these extra wires does no harm. The drawing below shows the minimum requirements that a cable must have to work properly with the P11.



Minimum Cable Requirements



DB9 to DB25 Cable Requirements



DB25 to DB25 Cable Requirements

6.3 Reset Push-Button Switch

Pressing the reset switch resets the target device. During normal programming(jumpers J3 & J4 positioned for the special bootstrap mode) a reset will place the target device in the special bootstrap mode and the WP11 software will prompt the operator when to press the reset button.

6.4 R1 Vpp Adjustment

Resistor R1 is used to adjust the Vpp voltage. The P11 board comes from the factory with the Vpp voltage set to 12.25 volts. The user should consult the specifications for the specific device being programmed to verify that this is the correct Vpp voltage. The Vpp voltage may be measured between TP2 and ground(ground can be obtained at the right hand end of capacitor C1).

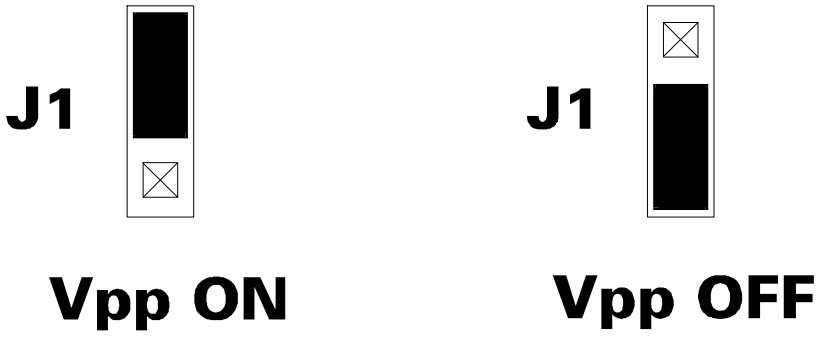
* * * WARNING * * *

Not all HC11 devices require an external Vpp voltage. Jumper J1 is used to turn off the Vpp voltage for those devices that do not require it. The WP11 software status panel shows whether or not Vpp should be on for the selected target device.

6.5 J1 Vpp Control Jumper

The position of the J1 jumper determines whether or not the external Vpp voltage is applied to the target device when power is applied to the P11 board. As stated above, **some HC11 devices do not require an external Vpp voltage** so please make sure that you have the J1 jumper set to the correct position for the specific target device being programmed.

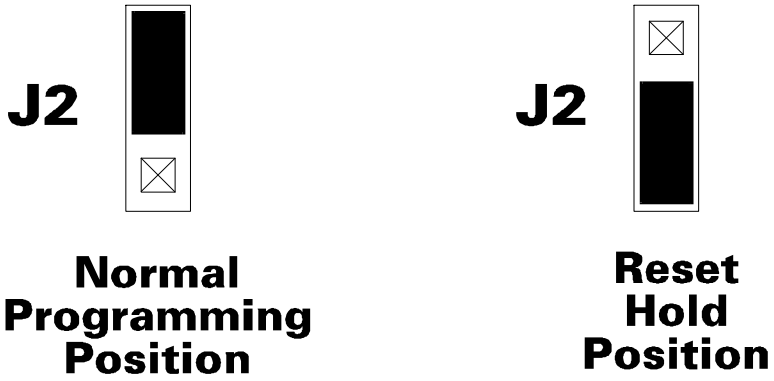
J1 Vpp Control Jumper Positions



6.6 J2 Reset Hold Jumper

P11 uses the special bootstrap mode for programming HC11 devices. If a secured device is brought out of reset in this mode the EEPROM memory of the device will be erased. *This may not be desirable!* The presence of this security feature can be detected while the target device is held in reset as described in section 6.7 below. The J2 jumper can be used to hold the target in reset while this test is performed.

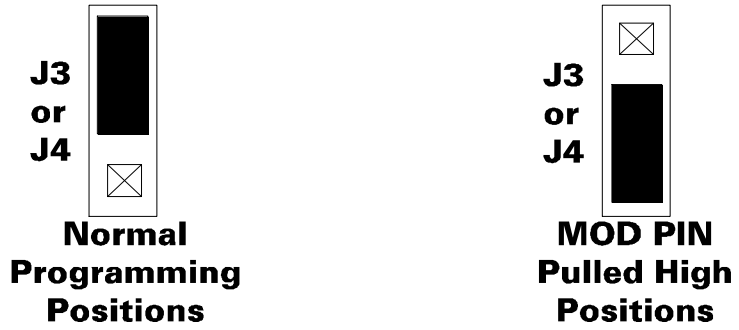
J2 Reset Hold Jumper Positions



6.7 J3 and J4 Mode Jumpers

These jumpers control the logic levels on the target device MODA and MODB pins. Normally these pins are held low for programming in the special bootstrap mode. If it is desired to test for the presence of the security option(determine if the security bit is in the secured state), these pins can be pulled high while the device is held in reset. If the security option is engaged, no signal will be present on TP1. If the security option is *not* engaged, the address strobe signal will be present on TP1.

J3 and J4 Mode Control Jumper Positions



6.8 ZIF Sockets

The PLCC ZIF sockets are Yamaichi auto-eject live bug sockets. To insert a chip into these sockets, place the chip right side up (so you can read the label on the chip and so that the pins of the chip are heading down into the socket) with the mitered corner of the chip matching the mitered corner of the socket, carefully align the chip and press down on the center of the chip until the chip goes into the socket and the outer frame of the socket raises up. To remove the chip, press down on the outer frame of the socket until the chip is ejected. If the P11 programmer board is oriented such that the mitered corner of the socket is the top left corner (approximately 11 o'clock), Pin 1 of the socket is at 12 o'clock.

The DIP ZIF sockets have PIN 1 located away from the locking handles.

*** * * NOTE * * ***

You must insert and remove devices from the ZIF sockets with power removed from the programming board.

7.0 THE WP11.EXE PROGRAM

WP11.EXE is a Windows program that is used to program 68HC11 microcontrollers by controlling hardware that supports the special bootstrap mode of these devices. WP11 uses a serial port to communicate with the programming hardware, which can be the P11 board, or any circuit that uses the HC11 special bootstrap mode.

WP11 is usually started from the "START" menu of your Windows PC or by double clicking on the WP11.EXE file name from within Windows Explorer.

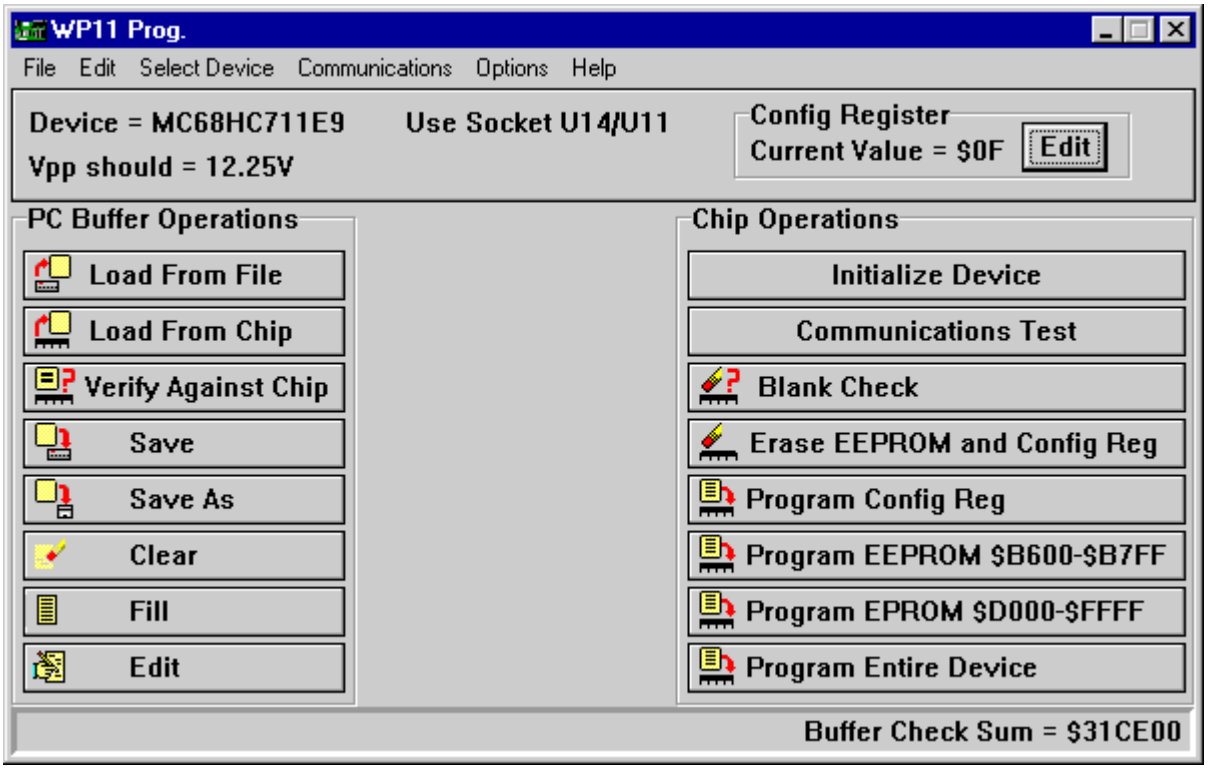
7.1 Program Overview

To program a device using WP11.EXE follow these steps:

1) Make sure that your PC serial port is properly set up to communicate with the hardware. Use the "Communications" menu item "Com Port Setup and Test" to verify proper operation of your serial port.
2) Select the desired HC11 family member using the "Select Device" menu.
3) Load the PC Buffer with the data that you want to place in the HC11 device. This data can come from a number of different sources, as we will see.
4) Make sure that the programming hardware is powered off and insert a device in the appropriate programming socket. Power up the programming hardware.
5) Click the "Initialize Device" button and follow the instructions.
6) Click on a programming button to perform the desired operation.

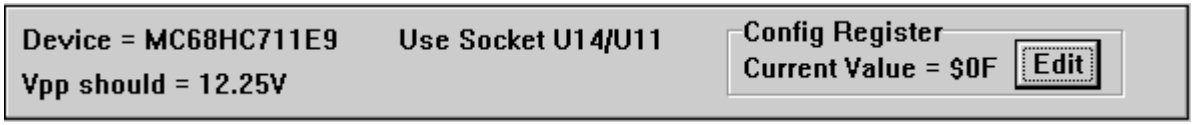
The rest of this section will explain the WP11 menu options.

7.2 WP11 Screen Shot



As you can see from the screen shot above WP11 has the usual Windows menu structure above a status panel. Below the status panel is a column of buttons on the left side, which are used to invoke various operations on the PC buffer. Remember, the PC buffer is memory inside your PC that contains the data that will be transferred to the HC11 chip during programming.
The column of buttons on the right side of the screen is used to invoke various chip programming operations.

7.2.1 WP11 Status Panel



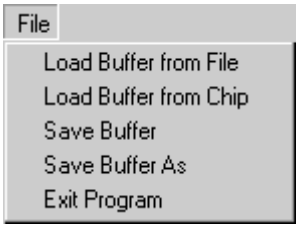
The status panel is used to show information about the currently selected device. It gets updated when a different HC11 device is chosen from the Select Device menu.

7.2.2 Changing the CONFIG Register

The Status Panel contains a means of changing the CONFIG register value. When the 68HC11 family member is changed the CONFIG Register value (in the PC) automatically changes to the the erased state of the new device. You may, however, wish to change the CONFIG Register to some other value. Pressing the "EDIT" button will enable you to specify a new value. You must be careful to specify values that are valid for the selected device. Changing the CONFIG Register will not affect the microcontroller until the device is programmed.

7.3 WP11 Menus

7.3.1 File Menu



Load Buffer from File is the same as ✍

Load Buffer from Chip is the same as ✍

Save Buffer is the same as ✍

Save Buffer As is the same as ✍

Load Buffer from File
Use this selection to load an object code file into the PC buffer.
When this option is selected, a standard Windows "Open" dialog box is displayed and the operator selects the desired object code file. File formats are automatically detected by WP11.EXE, and may be Motorola S-Record, Intel Hex or a binary memory image. Binary files must be 64K bytes in length. If the file name is entered correctly and can be opened and read by WP11.EXE, a memory image of the target chip is created in the buffer. If the file can't be opened, an error message is displayed and the operator is given another chance to enter the file name. This continues until the operator enters a correct file name or presses the "Cancel" button.

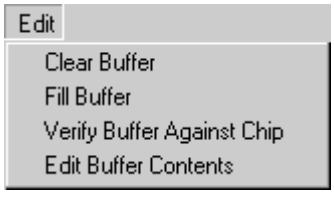
Load Buffer from Chip
Use this selection to load the contents of a programmed chip into the PC buffer.
A chip must be installed in the programming hardware and properly initialized for this menu option to be enabled. When this option is selected, a status Window with a Progress Gauge is displayed to keep the operator aware of the command progress. During the upload, the chip being read computes a checksum of the bytes sent to the PC. This checksum is sent to the PC where it is compared to a checksum of the received bytes computed by the PC. The data is placed in the buffer only if the correct number of bytes was received and the checksums match.

Save Buffer
This selection saves the contents of the buffer in a file on your PC. The file name used is the same as the currently opened file. If no file is currently opened a standard Windows "Save As" dialog box is opened to allow the operator to enter a file name. The file can be in one of three object code formats, Motorola S-Record, Intel HEX or binary image format. Use the "Options" menu to select the format.

Save Buffer As
This selection is very nearly the same as the "Save Buffer" menu item described above except the "Save As" dialog box is opened immediately so that a new file name can be specified.

Exit Program
This selection ends WP11.EXE execution. If the buffer was changed since the last save the operator is given the option of saving it. A WP11.INI file is created and saved in the WP11.EXE directory and is used to restore program options and settings the next time the program is used. The WP11.INI file is a standard text file and may be edited to change program options if need be. If WP11.INI cannot be found, WP11.EXE starts up with default settings for the program.

7.3.2 Edit Menu



Clear Buffer is the same as ✍

Fill Buffer is the same as ✍

Verify Buffer Against Chip is the same as ✍

Edit Buffer Contents is the same as ✍

Clear Buffer
This selection erases the data in the buffer in such a way as to leave each byte of the buffer the same as the corresponding byte in an erased chip. With presently supported devices each byte is returned to $FF. This function erases the current contents of the buffer.

Fill Buffer

This selection provides a means of filling the entire buffer with a byte of data that you specify. This function overwrites the current buffer contents. As an example of the use of this function consider this:

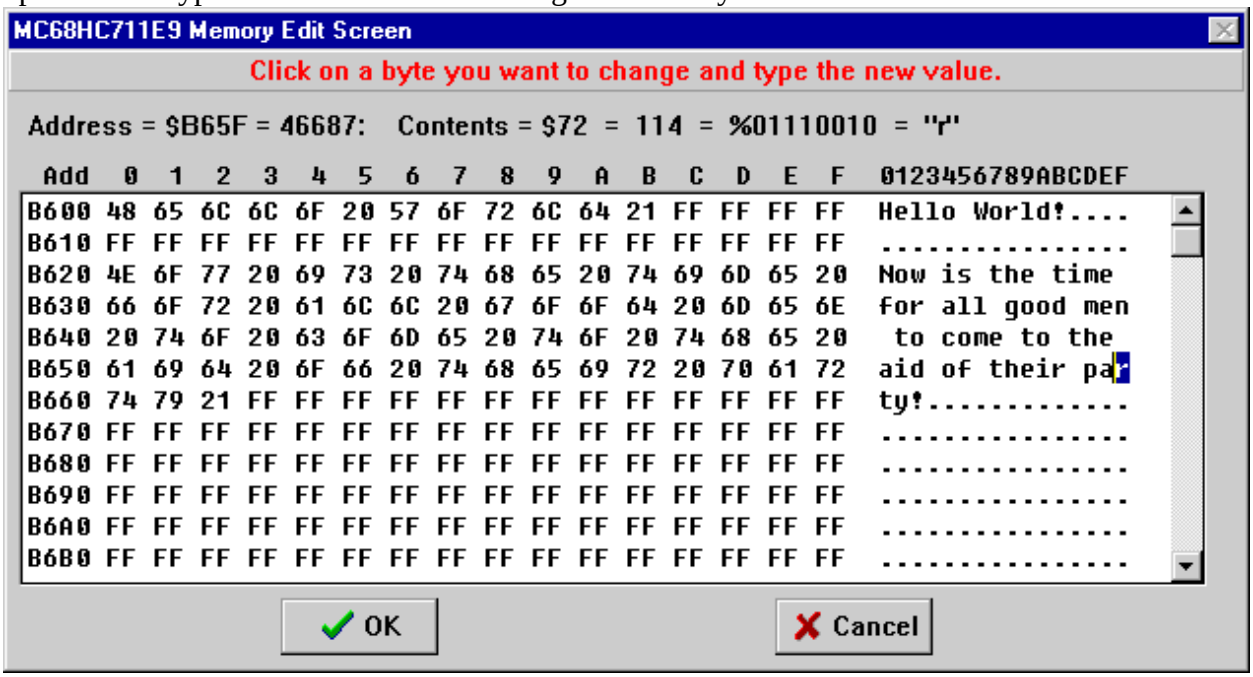
Suppose we want to have all the bytes in a device that are not taken up by our program set to a specific value. Say we want all unused memory locations filled with $3F, the op-code for the SWI instruction. To do this use the "Fill Buffer" function and specify $3F as the fill byte. Then use the "Load Buffer from File" function to load your program into the buffer. The buffer is not changed before the "Load Buffer from File" function is executed and the "Load Buffer from File" function does not change any bytes in the buffer that do not have data in the object code file. We started with the buffer completely filled with $3F and the "Load Buffer from File" function overwrote only those bytes that had data specified in the object code file so we end up with all of the bytes in the buffer that are not used by our program still set to $3F.

Verify Buffer Against Chip

This selection first uploads data from the chip similar to the "Load Buffer from Chip" function. The same dialog box and progress gauge is used during the upload. If uploading of the data is successful then each byte of the uploaded data is compared to the corresponding byte in the buffer. Data in the buffer is not changed. A pass/fail dialog box is displayed to end the function

Edit Buffer Contents

When this function is selected data in the buffer is copied to a special edit area. The contents of this special edit area are then displayed on a Memory Edit Screen as shown below where the operator can type new data over the existing data directly on the screen.

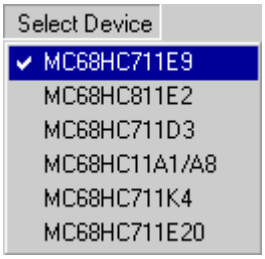

MC68HC711E9 Memory Edit Screen
Click on a byte you want to change and type the new value.
Address = $B65F = 46687: Contents = $72 = 114 = %01110010 = "r"
Add 0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
B600 48 65 6C 6C 6F 20 57 6F 72 6C 64 21 FF FF FF FF Hello World!....
B610 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
B620 4E 6F 77 20 69 73 20 74 68 65 20 74 69 6D 65 20 Now is the time
B630 66 6F 72 20 61 6C 6C 20 67 6F 6F 64 20 6D 65 6E for all good men
B640 20 74 6F 20 63 6F 6D 65 20 74 6F 20 74 68 65 20 to come to the
B650 61 69 64 20 6F 66 20 74 68 65 69 72 20 70 61 72 aid of their pa
B660 74 79 21 FF FF FF FF FF FF FF FF FF FF FF FF FF ty!.............
B670 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
B680 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
B690 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
B6A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
B6B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ....
OK
Cancel

Each line in the edit box above shows 16 bytes of the special edit area data in both hex and ASCII formats with the address of the first of the 16 bytes shown at the beginning of each line.

As an example, in the sixth line shown above, the ‘r’ in “party” is highlighted. We just clicked on the ‘r’ to highlight it. The status line at the top of the screen tells us that the ‘r’ is at address $B65F. It further tells us that $B65F = 46687 decimal and that the hex value of the ‘r’ is $72 which is 114 decimal which is 01110010 binary. If we wanted to change the ‘r’ we would simply type the new value and the screen would be updated and the cursor would move to highlight the next character. We can also edit in the HEX display area by simply clicking on the character we want to change and typing the new value. It’s all very easy. Try it!

Clicking the “OK” button copies the data in the special edit area back to the buffer thereby making the edits take effect. Clicking the “Cancel” button discards the data in the special edit area leaving the buffer unchanged.

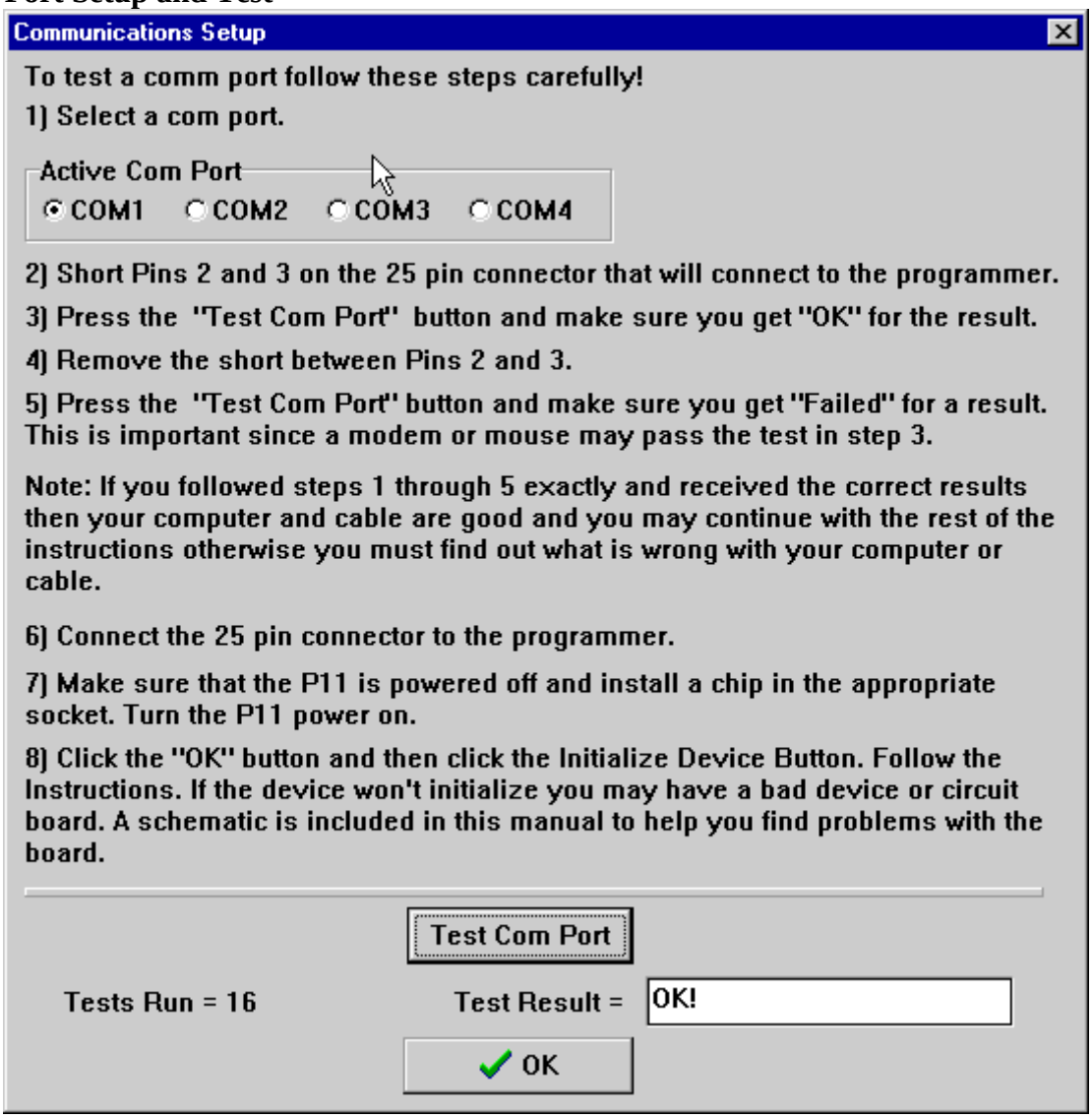
7.3.3 Select Device Menu



This menu is used to select the HC11 family member that you want to program. The part number is checked when it is the currently selected device. The part number is also displayed on the status panel with the “Device = XXXXXXXXXX” label.

7.3.4 Communications Menu

Com Port Setup and Test


Communications Setup
To test a comm port follow these steps carefully!
1) Select a com port.
Active Com Port
COM1
COM2
COM3
COM4
2) Short Pins 2 and 3 on the 25 pin connector that will connect to the programmer.
3) Press the "Test Com Port" button and make sure you get "OK" for the result.
4) Remove the short between Pins 2 and 3.
5) Press the "Test Com Port" button and make sure you get "Failed" for a result. This is important since a modem or mouse may pass the test in step 3.
Note: If you followed steps 1 through 5 exactly and received the correct results then your computer and cable are good and you may continue with the rest of the instructions otherwise you must find out what is wrong with your computer or cable.
6) Connect the 25 pin connector to the programmer.
7) Make sure that the P11 is powered off and install a chip in the appropriate socket. Turn the P11 power on.
8) Click the "OK" button and then click the Initialize Device Button. Follow the Instructions. If the device won't initialize you may have a bad device or circuit board. A schematic is included in this manual to help you find problems with the board.
Test Com Port
Tests Run = 16
Test Result =
OK!
OK

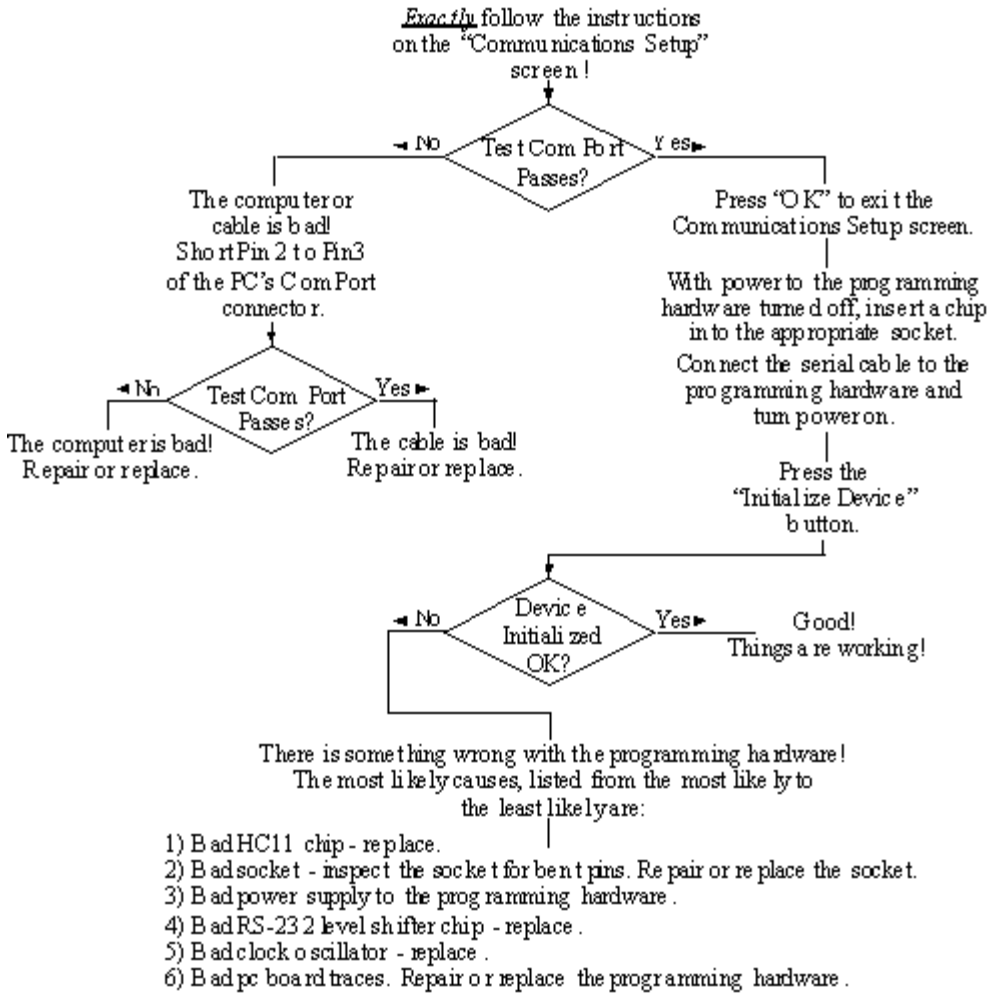
When the Com Port Setup and Test menu item is selected the Communications Setup screen shown above is displayed. The screen has detailed instructions and a test facility for determining whether or not a working serial port and cable are in use.

*** * * Important Note * * ***

In order to perform any of the chip operations the computer must be able to communicate with the programming hardware. Our experience indicates that the most difficult task faced by a first time user is to establish this communication. This has to do with the non-standard nature of the EIA RS-232 Standard and the many different RS-232 cables that exist. If communications aren't working, the ***very first*** thing to do is to determine if the cause lies with the computer and cable or with the programming hardware.

The test facility and instructions on this screen provide the tools required to determine if the problem is with the computer and cable or with the programming hardware.

Com Port Troubleshooting Chart


Exactly follow the instructions on the "Communications Setup" screen!
Test Com Port Passes?
No
Yes
The computer or cable is bad! Short Pin 2 to Pin3 of the PC's Com Port connector.
Test Com Port Passes?
No
Yes
The computer is bad! Repair or replace.
The cable is bad! Repair or replace.
Press "OK" to exit the Communications Setup screen.
With power to the programming hardware turned off, insert a chip into the appropriate socket.
Connect the serial cable to the programming hardware and turn power on.
Press the "Initialize Device" button.
Device Initialized OK?
No
Yes
Good! Things are working!
There is something wrong with the programming hardware! The most likely causes, listed from the most likely to the least likely are:
1) Bad HC11 chip - replace.
2) Bad socket - inspect the socket for bent pins. Repair or replace the socket.
3) Bad power supply to the programming hardware.
4) Bad RS-232 level shifter chip - replace.
5) Bad clock oscillator - replace.
6) Bad pc board traces. Repair or replace the programming hardware.

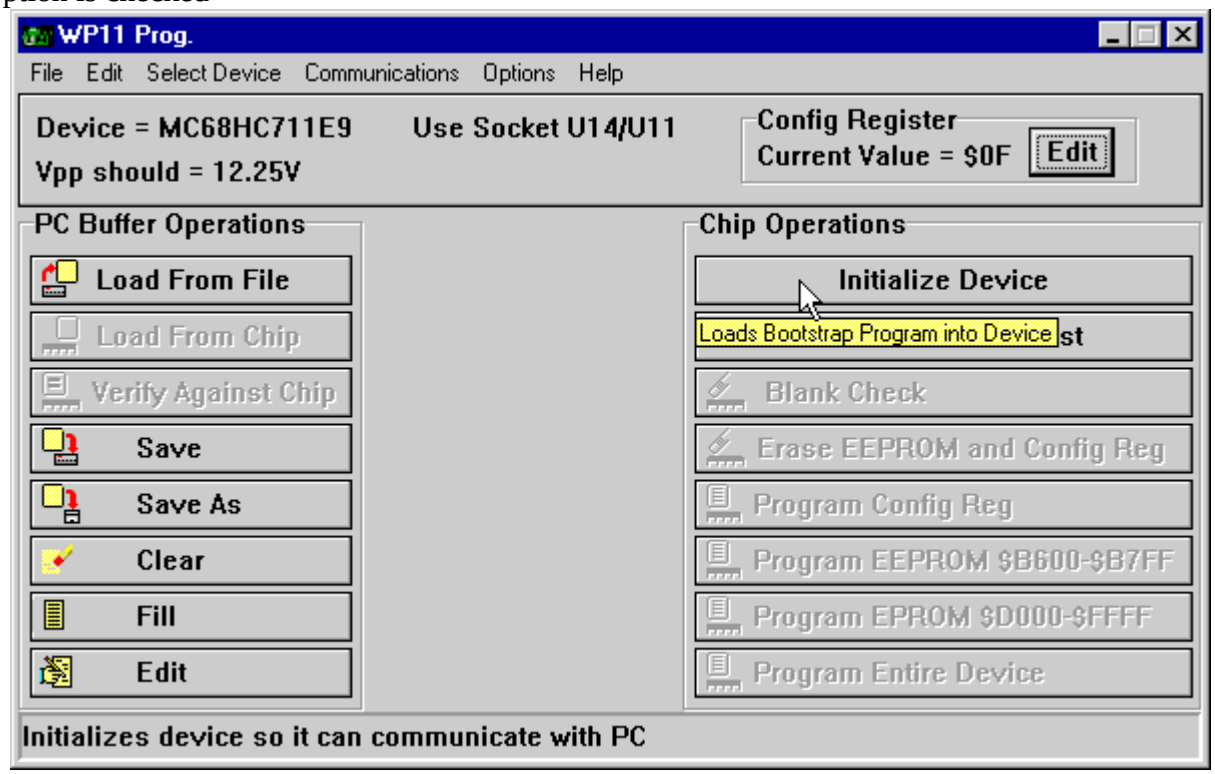
7.3.5 Options Menu

Options
File Format = Motorola S19
File Format = Intel Hex
File Format = Binary
Load Buffer On Startup
Turn Off Hint Boxes
Turn Off Status Line Hints
Turn Off Memory Map Checks

File Format Options
The first three Option Menu choices allow the user to specify the format to use when saving the buffer in an object code file. This format will be used when the "Save" or "Save As" buttons are pressed.
Note: When loading a file into the buffer the file format is automatically detected so these options come into play only when saving the buffer contents to a file.

Load Buffer On Startup
WP11 remembers the file name of the object code file loaded or saved in an ".INI" file. When the "Load Buffer On Startup" option is checked WP11 automatically loads this file back into the buffer.

Turn Off Hint Boxes
Referring to the screen shot below notice that the cursor is located over the "Initialize Device" button. Also notice that just below the cursor is a box containing the text "Loads Bootstrap Program into Device". This box is called a "hint box" and tells the operator what will happen when the button is pressed. These fly over hint boxes are probably not needed once a user is familiar with the program. The hint boxes will not be displayed when the "Turn off Hint Boxes" option is checked



Turn Off Status Line Hints

Referring to the screen shot above notice that the cursor is located over the "Initialize Device" button and that the panel at the bottom of the WP11 window has the text "Initializes device so it can communicate with PC". This text is called a "help hint" and gives the user more information about what will happen when the "Initialize Device" button is pressed. These fly over help hints are probably not needed once a user is familiar with the program. The help hints will not be displayed when the "Turn off Status Line Hints" option is checked.

Turn Off Memory Map Checks

WP11 knows the memory maps of supported family members. That is to say that it knows where in memory the EEPROM, EPROM , CONFIG Register and vectors are located. It knows which memory locations are valid for a particular family member and which ones are invalid. Normally, when WP11 is loading the buffer with data in an object code file it checks each byte to verify that it is going into a valid memory location for the currently selected device. If errors are detected they are reported to the operator. Generally, this is just what we want because if, by some mistake, the program doesn't fit in the device memory map the program won't run properly. But, what if all we want to do is load an object code file into the buffer, perhaps edit it, and then save it in a different file or file format? Having this normally nice feature turned on might prevent us from doing this simple thing. To solve this problem, check the "Turn Off Memory Map Checks" option.

7.3.6 Help Menu

Contents

Selecting "Help – Contents" opens the WP11 help file.

About

Selecting "Help – About" opens the WP11 About screen where information about the program version number and our company contact information can be found.

7.4 Chip Operation Buttons

Initialize Device

WP11 uses the HC11 special bootstrap mode to program devices. This requires that a small program be downloaded to the target device RAM for the purpose of controlling the programming process and communicating with the PC. The "Initialize Device" button performs this task. No programming operation on a chip can be done without initializing the chip. This is why all menu items and buttons that talk to the programming hardware are disabled until the chip is initialized.

The small programs that "Initialize Device" downloads to the target RAM ***must*** be in the same directory as the WP11.EXE program so that they can be found when needed. The programs are different for each family member and have names such as "P11_E9.MIK" for the

MC68HC711E9 chip and “P11_K4.MIK” for the MC68HC711K4 chip. If WP11 can’t find these files an error message is generated.

When the “Initialize Device” button is pressed a dialog box is displayed that has a progress gauge to show the status of the operation. If the operation is successful the dialog box just goes away and the appropriate menu items and buttons are enabled. If the operation is not successful an error message is generated.

Normally this initialization process has to be performed only once when a target device is inserted into the programming hardware. It does *not* have to be repeated unless power is turned off to the target device or the target device is reset.

Communications Check

This button is used to determine if the target device is properly initialized. The PC sends $00 to the programming hardware and expects to receive $AA followed by $55. If the expected response is received then the appropriate menu items and buttons are enabled else they are disabled.

The “Communications Test” button can be used to enable the chip operations buttons when you know that the target device is properly initialized but the button are not enabled such as would be the case if you exited and re-entered the program. Of course, you could use the “Initialize Device” to do the same thing but it would take much longer.

Blank Check

Press this button to test the target device's CONFIG register, EPROM and EEPROM memory to determine if every location is in the erased state.

Erase EEPROM and Config Reg

This button returns the EEPROM and CONFIG register in the target device to its erased state.

Program CONFIG Reg

Pressing this button programs the value shown for the CONFIG register in the WP11 status panel into the CONFIG register of the target device then tests to determine if the operation was successful. This button will not be enabled if the currently selected device does not have a CONFIG register.

Program EEPROM

Pressing this button programs the contents of the EEPROM portion of the WP11 buffer into the EEPROM memory of the target device then tests to determine if the operation was successful.

Program EPROM

Pressing this button programs the contents of the EPROM portion of the WP11 buffer into the EPROM memory of the target device then tests to determine if the operation was successful.

Program Entire Device

Pressing this button programs the contents of the EEPROM & EPROM portion of the WP11 buffer and the value shown for the CONFIG register on the WP11 status panel into the target device then tests to determine if the operation was successful.

8.0 TUTORIAL EXAMPLE SESSION

Now that you have tested the P11 Programmer and it is functioning correctly, please invest the time to read and follow through this tutorial. Two demonstration/test 68HC11 programs were included for this purpose. Going through the tutorial session will take just a short time and will provide you with an overview of the P11 by actually using it.

It is assumed that you have read the rest of this manual, installed the software in accordance with the *Getting Started* section and performed the *Initial Test of the Programmer* in accordance with that section of this manual. It is further assumed that you have invoked WP11.EXE.
It is also assumed that you have a MC68HC711E9 or a MC68HC811E2 chip that can be used with this tutorial. If you are using an E9 chip, we are not going to program the EPROM section, we will only program the EEPROM section so the chip can be erased and re-used for another purpose.

The steps required to program a chip are as follows:

1) Select the desired device type from the *Select Device* menu.
2) Insert the target device into the proper socket. **NOTE:** You must insert and remove devices from the ZIF sockets with power removed from the programming board.
3) Load the *Buffer* with the data that you want to program into the target device. This can be done in a number of ways, from a file, from another chip, manually etc.
4) Initialize the target device for programming.
5) Program the target device using one of the available *chip operations* buttons.

Tutorial:

1) Load the Buffer with the desired data to be programmed into the target device by pressing the *Load From File* button on the WP11 main screen. From the Open dialog box open "TEST_E9.MIK" if you are using an E9 or "TEST_E2.MIK" if you are using an E2 target device.
2) Press the *Edit* button and use the scroll bars to view the data. Exit the screen by pressing the *Cancel* button.

3) Place the target device in the correct socket which is shown on the status panel. **NOTE:** You must insert and remove devices from the ZIF sockets with power removed from the programming board.
4) Press the *Initialize Device* button. This downloads a small program to the target device that controls the programming process.c
5) Press the *Program & Verify EEPROM* button if using an E9 device or *Program & Verify Device* if using an E2 device. This programs the contents of the *Buffer* into the target device.
6) To prove that we can read the contents of a programmed device, press the *Clear* button to clear the buffer, verify this with the edit button if you are the untrusting type, then press the *Load From Chip* button. Then press the *Edit* button. You should see the data that was originally programmed into the device.

That's it! You have just performed all of the necessary steps to program an HC11 device.

We sincerely hope that after using this tool, you will consider it to be a sound investment. We look forward to your comments and suggestions and to providing any assistance with their use that you may require.

9.0 USING THE P11.EXE PROGRAM

All interactions with the programming hardware are accomplished by selecting the various options provided by the P11.EXE program.

9.1 Invoking P11.EXE

P11.EXE is invoked by typing at the DOS prompt:

```
>P11<Enter>
```

Assuming the use of a hard disk and the use of the PATH command in the AUTOEXEC.BAT you may invoke P11.EXE from any directory.

If there are multiple projects or multiple versions of the same project that require different settings, then you should create a different directory for each project or version. If your project or version requires the use of a different microcontroller, you should start P11.EXE from within the directory of that project or version because, when P11.EXE is exited, a file called P11.INI is created to save the current settings and this file is saved in the current directory. Whenever you invoke the program from that directory again, the P11.INI file determines the settings the program starts with. Any time P11.EXE is invoked from a directory that does not contain a P11.INI file, the default settings for P11.EXE (shown below in Section 6.2) will be in place.

Assuming the use of P11.EXE from a bootable working floppy disk with one project per disk, you may invoke P11.EXE from the A:\ or B:\ prompt, whichever contains the floppy disk..

9.1.1 Command Line Options

Command line options take precedence over settings in the P11.INI file.

From the Command Line the user may change the Communications Port that P11.EXE starts up with to either **COM1** or **COM2**. The default is COM1. The communication port setting is saved in the P11.INI file found in the directory that P11.EXE was last started up in.

For example:

```
C:\HC11PROJ\>p11 com1<Enter>
```

will start P11.EXE on COM Port 1. The COM port may also be set from the P11.EXE main menu.

9.1.2 Command Line Only Options

There are two program options of `P11.EXE` that can *only* be modified from the Command Line. These parameters are the **Video Mode** and the **Program Beep**.
For most circumstances the `P11.EXE` video is auto-sensing. However for some configurations, notably for computers with older color video cards that have pseudo color monitors attached, the video mode may be unacceptable.

BW80 ;sets the video mode to Black & White, 80 columns

Example:

```
C:\HC11PROJ\>p11 BW80<Enter>
```

will start `P11.EXE` in Black & White video mode. Video modes are not saved to the `P11.INI` file.

There is an optional **Program Beep** for those who wish it. It sounds whenever a task is completed. This option is saved in the `P11.INI` file.

It may be set by typing:

```
C:\HC11PROJ\>p11 beep<Enter>
```

when beginning a `P11.EXE` session. To return to silent operation, type:

```
C:\HC11PROJ\>p11 nobeep<Enter>
```

at the command line when beginning the next `P11.EXE` session.
Command line options may be in any order:

Example:

```
C:\HC11PROJ\>p11 nobeep BW80<Enter>
```

will turn the beep off and set the video to Black & White 80 columns.

9.2 P11.EXE Screen Display

When `P11.EXE` is invoked, the program menu screen is displayed. It looks something like this:

This is the default setting for the `P11.EXE`. Before describing each of the menu options some words on the general operation of the program are in order.

`P11.EXE` maintains an exact image of the microcontroller's memory in the PC. This memory image is called the *buffer*. During programming operations, data from the *buffer* is copied (programmed) into the memory of the microcontroller device. The *buffer* must be loaded with the desired data before a programming operation is performed. The *buffer* can be loaded from an object code file or from another microcontroller that already contains the desired data. The *buffer* can be displayed and edited before a programming operation is performed. With this general understanding, each of `P11.EXE`'s options will now be described.

9.3 P11.EXE Menu Options

Change Device Type

Selecting the **Change Device Type** option brings up the device type selection menu that looks like this:

With the up and down arrow keys, select the processor desired and press `<Enter>`.

* * * WARNING * * *

Changing the device type or even selecting the same device type as the current selection purges the contents of the buffer. The buffer is filled with the erased state of the selected device.

Change Buffer Fill Character

When you change the 68HC11 device type the buffer fill character automatically changes to the erased state of the selected device. You may, however, wish to change the buffer fill character to some other value. Pressing `<Enter>` at this **Main Menu** selection will enable you to specify a HEX value between 0H & 0FFH inclusive. Changing the buffer fill character will completely overwrite the buffer.

Change Config Register

When you change the 68HC11 device type the Config Register value (in the PC) automatically changes to the erased state of the selected device. You may, however, wish to change the Config Register to some other value. Pressing `<Enter>` at this **Main Menu** selection will enable you to specify a HEX value between 0H & 0FFH inclusive. You must be careful to specify values for the Config Register that are valid for the selected device. Changing the Config Register in the Main Menu will not effect the microcontroller until the Config Register is programmed from the Device Menu.

Load Object Code File to Buffer

This command is used to load an object code file into the host PC's RAM buffer.
When this option is selected, the operator is prompted to enter the file name of the object code file. The complete file name must be entered including the file extension. File formats are automatically detected by `P11.EXE`, and may be either Motorola S-Record, Intel Hex or a binary memory image. Binary files must be 64K bytes in length. If the file name is entered correctly and can be opened and read by `P11.EXE`, a memory image of the target chip is created in the buffer. If the file can not be opened, an error message is displayed and the operator is given another chance to enter the file name. This continues until the operator enters a correct file name or presses `<Esc>`. The operator is then given the option of exiting the screen.

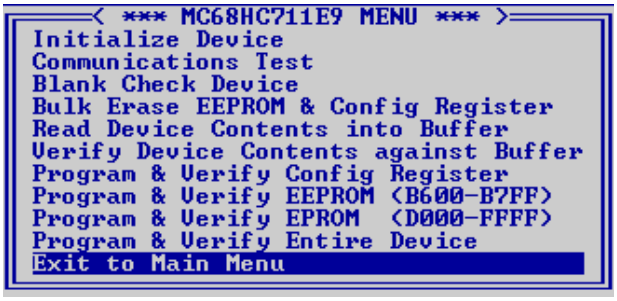
You may then **Examine/Edit Buffer Contents**. For a complete description of this option see below.

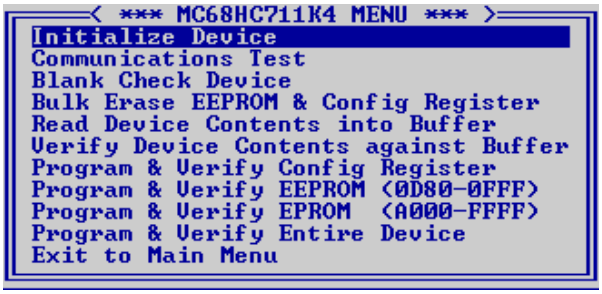
* * * WARNING * * *

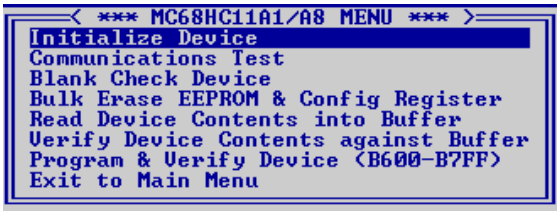
The order in which you use the commands from the main menu is very important. You should select the desired HC11 device before you fill the buffer with programming data. When you change the device type (or even select the same device type from the **Change 68HC11 Device Type** menu) the buffer is filled with the erased state of the selected device. Next if you wish to put a different fill character in the buffer you must do so before you **Load Object Code File to Buffer**. If you load the object code file first, then change the fill character, the fill character will completely overwrite the contents of the buffer erasing any information currently in it. After selecting the device type and changing the fill character, the object code file may be loaded.

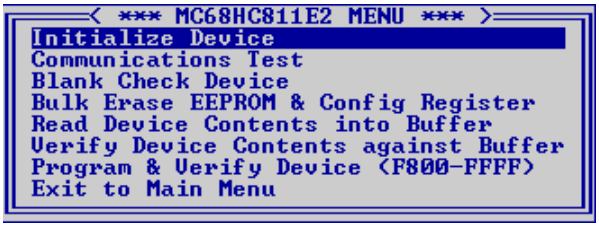
Device Menu

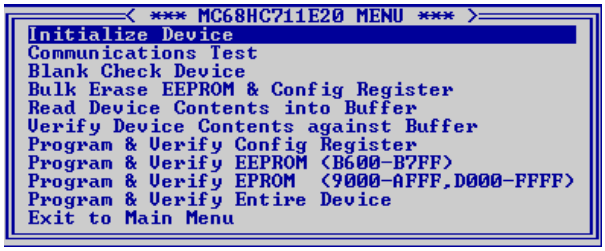
These sub menus let you select various options for each respective selected device.


*** MC68HC711E9 MENU ***
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (B600-B7FF)
Program & Verify EPROM (D000-FFFF)
Program & Verify Entire Device
Exit to Main Menu


*** MC68HC711K4 MENU ***
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (0D80-0FFF)
Program & Verify EPROM (A000-FFFF)
Program & Verify Entire Device
Exit to Main Menu


*** MC68HC11A1/A8 MENU ***
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Device (B600-B7FF)
Exit to Main Menu


*** MC68HC811E2 MENU ***
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Device (F800-FFFF)
Exit to Main Menu


*** MC68HC711E20 MENU ***
Initialize Device
Communications Test
Blank Check Device
Bulk Erase EEPROM & Config Register
Read Device Contents into Buffer
Verify Device Contents against Buffer
Program & Verify Config Register
Program & Verify EEPROM (B600-B7FF)
Program & Verify EPROM (9000-AFFF,D000-FFFF)
Program & Verify Entire Device
Exit to Main Menu

For each command option there is a text screen displayed on the host computer. The user must follow these instructions and wait for the completion of the command.

Initialize Device

P11 uses the HC11 special bootstrap mode during programming. This mode requires that special programming code be downloaded to the target device's RAM to control the programming process. The Initialize Device menu option performs this task. No chip may be programmed or communicated with successfully without first performing this menu function. Normally, this initialization process has to be performed only once after the target device is brought out of reset. It does *not* have to be repeated unless power is turned off to the target device.

Communications Test

This command is used to determine if the target device is properly initialized.

Blank Check Device

Tests the target device's Config Register, EPROM and EEPROM memory to determine if every location is in the erased state.

Bulk Erase EEPROM & Config Reg.
This menu command returns the EEPROM and Config Register in the target device to its erased state.

Read Device Contents into Buffer
This command reads the contents of the target device's programmable memory into the RAM buffer in the host computer.

Verify Device Contents against Buffer
This command compares the contents of the target device's programmable memory with the contents of the P11 buffer.

Program & Verify Config Register
This command programs the value of the Config Register shown on the P11 **status line** into the Config Register of the target device then tests to determine if the operation was successful.

Program & Verify EEPROM
This command programs the contents of the EEPROM portion of the P11 buffer into the EEPROM memory of the target device then tests to determine if the operation was successful.

Program & Verify EPROM
This command programs the contents of the EPROM portion of the P11 buffer into the EPROM memory of the target device then tests to determine if the operation was successful.

Program & Verify Entire Device
This command programs the contents of the EEPROM & EPROM portion of the P11 buffer and the value of the Config Register shown on the P11 **Status Line** into the target device then tests to determine if the operation was successful.

Exit to Main Menu
This option returns the user to the `P11.EXE` Main Menu.

This is the end of the description of the **Device Menu** sub menu. We will now return to a description of the remaining selections in the **Main Menu**.

Examine/Edit Buffer Contents
Selecting this option allows the user to examine and/or edit the contents of the P11 buffer.

If the user wishes, the buffer may be edited by typing over the current contents of memory locations. The cursor may be moved around the screen using the arrow keys. Pressing the TAB key toggles the cursor between the HEX portion of the screen and the ASCII portion of the screen. Non-HEX keys are disabled in the Hex portion of the screen, non-printable characters are disabled in the ASCII portion of the screen.

To move through the memory map in the buffer, use the Page Up and Page Down keys. Each page displays 256 bytes of information. The buffer varies in size according to the currently selected device.

Save Buffer in a File

This option allows you to save the current contents of the P11 buffer in one of three object code file formats, Motorola S-Record, Intel HEX or binary image format.

When you select this option, you are asked to name the new file. If you wish to save the file in a directory other than the current directory you must type the full path and file name.

Change Communications Port

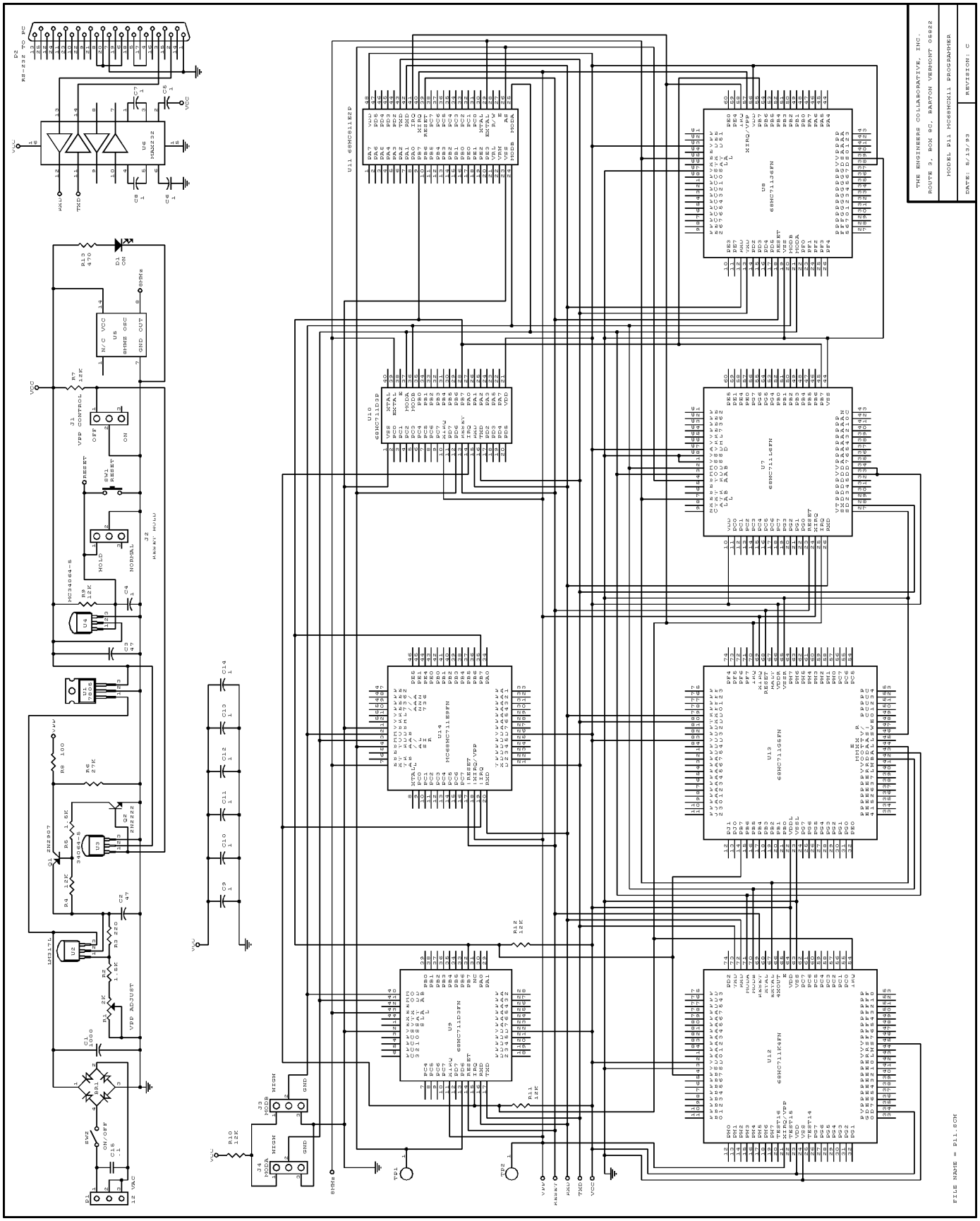
This option toggles the active communications port between COM 1 and COM 2. The currently active port is displayed on the left side of the P11 status line.

Help

This option opens the help screen. The help screen may be scrolled through using the `Page Up` and `Page Down` keys or using the arrow keys. To exit the Help screen, press <`Esc`>.
The help screen information is contained in a standard DOS text file. The user may edit this file to add any additional information that may be required.

Exit to DOS

This menu selection will end the `P11.EXE` session. Pressing <End> will take you to this menu selection from anywhere else in the **Main Menu**.


THE ENGINEERS COLLABORATIVE, INC.
ROUTE 3, BOX 8C, BARTON VERMONT 05822
MODEL P11 MC68HC811 PROGRAMMER
DATE: 8/13/93
REVISION: C
FILE NAME = P11.SCH